Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) An apparatus for performing exponentiations, comprising:

a set of computational devices containing first and second subsets, wherein the first subset has a plurality of members which are chained together such that the devices of the first subset can operate both independently and as members of a first computational chain, and wherein the second subset has a plurality of members which are chained together such that the devices of the second subset can operate both independently and as members of a second computational chain distinct from said first computational chain; and

a chaining controller adapted to instruct the first subset of devices to act as a first computational chain when the apparatus is required to perform an exponentiation of a first size, and being further adapted to instruct the second subset of devices to act as a second computational chain when the apparatus is required to perform an exponentiation of a second size distinct from said first size;

wherein each computational device is adapted to perform 1024-bit exponentiation, wherein said chaining controller is adapted to instruct the first subset of devices to act as a first computational chain when the apparatus is required to perform 2048-bit exponentiation, and wherein said chaining controller is adapted to instruct the second subset of devices to act as a second computational chain when the apparatus is required to perform 4096-bit exponentiation.

2. (Cancelled) The apparatus of Claim 1, wherein said computational devices are exponentiators, and wherein the first computational chain comprises a first exponentiation chain.

3. (Currently Amended) The apparatus of Claim 1 ~~Claim 2~~, further comprising a hardware state controller for each computational device ~~exponentiator~~ of the first computational ~~exponentiation~~ chain, wherein each hardware state controller includes replicated fanout control logic.

4. (Currently Amended) The apparatus of Claim 3, wherein the replicated fanout control logic is configured to allow computational devices ~~exponentiators~~ of the first computational ~~exponentiation~~ chain to chain without delay due to high fanout.

5. (Currently Amended) The apparatus of Claim 3, wherein the replicated fanout control logic is configured such that state machines of the first computational ~~exponentiation~~ chain sequence efficiently.

6. (Currently Amended) The apparatus of Claim 1 ~~Claim 2~~, wherein each computational device ~~exponentiator~~ further comprises a custom multiplier datapath, ~~datapath~~; and wherein each custom multiplier datapath is configured so that the length of its longest wire is short.

7. (Currently Amended) The apparatus of Claim 6, wherein the custom multiplier datapaths of chained computational devices ~~exponentiators~~ are physically mirrored to each other so that the wire length between the two is short.

8. (Previously Presented) The apparatus of Claim 6, wherein the custom multiplier datapath has a serpentine layout so that the wire length between the most separated adjacent data locations is short.

9. (Currently Amended) The apparatus of Claim 1 ~~Claim 2~~, wherein the number of computational devices ~~exponentiators~~ in the ~~plurality of~~ set of computational devices ~~exponentiators~~ equals 2^k, and wherein k is a non-negative integer.

10. (Previously Presented) The apparatus of Claim 9, wherein k equals 2.

11. (Currently Amended) The apparatus of Claim 9, wherein each computational device ~~exponentiator~~ is adapted to exponentiate a 512-bit number.

12. (Currently Amended) The apparatus of Claim 1 ~~Claim 2~~, wherein the number of computational devices ~~exponentiators~~ in the first computational ~~exponentiation~~ chain equals 2^k wherein k is a positive integer.

13. (Currently Amended) The apparatus of Claim 12, further comprising:

a second computational ~~exponentiation~~ chain;

wherein the second subset includes at least two computational devices ~~exponentiators~~, wherein the chaining controller is configured to instruct the second subset to operate as a second computational ~~exponentiation~~ chain, and wherein no computational device ~~exponentiator~~ of the first computational ~~exponentiation~~ chain is part of the second computational ~~exponentiation~~ chain.

14. (Currently Amended) The apparatus of Claim 1 ~~Claim 2~~, wherein each computational device in said set of computational devices ~~exponentiator~~ further comprises:

a cleave/merge engine;

wherein the cleave/merge engine is configured to:

receive AA, which is a 2w-bit number;

calculate A_1 and A_2, which are two w-bit numbers based on AA; and

output A_1 and A_2;

wherein the cleave/merge engine is also configured to:

receive B_1 and B_2, which are two w-bit numbers;

calculate BB, which is a 2w-bit number based on B_1 and B_2; and

output BB;

wherein exponentiation of AA yields BB;

wherein exponentiation of A_1 yields B_1;

wherein exponentiation of A_2 yields B_2; and

wherein w is a positive integer.

15. (Previously Presented) The apparatus of Claim 14, wherein A_1 and A_2 are calculated from AA, and BB is calculated from B_1 and B_2, using a scalable Chinese Remainder Theorem implementation.

16. (Currently Amended) The apparatus of Claim 15,

wherein each computational device ~~exponentiator~~ is adapted to perform 1024-bit exponentiation;

wherein, if 2048-bit exponentiation is required, the chaining controller causes the first computational ~~exponentiation~~ chain to comprise two exponentiators; and

wherein, if 4096-bit exponentiation is required, the chaining controller causes the first computational ~~exponentiation~~ chain to comprise four exponentiators.

17. (Cancelled) A system for computing comprising:

a computing device;

at least one apparatus of Claim 1; and

wherein the computing device is configured to use the apparatus of Claim 1 to perform computations.

18. (Previously Presented) A method for encrypting/decrypting data comprising:

providing a 1024-bit number X; and

encrypting/decrypting X by

loading X into session memory,

cleaving X mod P to compute X_P,

cleaving X mod Q to compute X_Q,

exponentiating X_P to compute C_P,

exponentiating X_Q to compute C_Q,

merging C_P and C_Q to compute C, wherein C is a 1024-bit number, and

retrieving C from the session memory;

wherein the method further comprises (a) selecting one of a plurality of session controllers; (b) setting the busy bit for the one session controller; and (c) clearing the busy bit for the one session controller.

19. (Previously Presented) The method of Claim 18, further comprising:
selecting one session controller of 32 available session controllers;
setting the busy bit for the one session controller;
wherein the argument X is a 1024-bit number;
wherein C is a 1024-bit number; and
clearing the busy bit for the one session controller.

20. (Previously Presented) The method of Claim 18, further comprising:
selecting two session controllers of 32 available session controllers;
setting the busy bits for the two session controllers
wherein loading argument X into session memory includes:
loading part of the argument X into the session memory of one of the two session controllers;
loading the remainder of the argument X into the session memory of the other of the two session controllers;
wherein the argument X is a 2048-bit number;
wherein C is a 2048-bit number; and
clearing the busy bits for the two session controllers.

21. (Previously Presented) The method of Claim 18, further comprising:
selecting four session controllers of 32 available session controllers;
setting the busy bits for the four session controllers
wherein loading argument X into session memory includes:
loading a first part of the argument X into the session memory of a first of the four session controllers;
loading a second part of the argument X into the session memory of a

second of the four session controllers;

loading a third part of the argument X into the session memory of a third of the four session controllers;

loading the remaining of the argument X into the session memory of a fourth of the four session controllers;

wherein the argument X is a 4096-bit number;

wherein C is a 4096-bit number; and

clearing the busy bits for the four session controllers.

22. (Previously Presented) The method of Claim 18,

wherein the cleaving X mod P comprises:

setting A[513:0] = X[1023:51O];

calculating Z[1026:0] = A[513:0] x μP[512:0], wherein μP[512] = 1;

setting B[513:0] = Z[1026:512];

setting C[513:0] = X[513:0];

calculating Y[1025:0] = B[513:0] x P[511:0];

setting D[513:0] = Y[513:0];

calculating E[513:0] = C[513:0] - D[513:0];

if E > P then calculating E = E - P;

if E > P then E = E - P; and

setting X_P = E[511 :0] as the result of the cleaving X mod P, whereby X_P equals X mod P; and

wherein the cleaving X mod Q comprises:

setting A[513:0] = X[1023:510];

calculating Z[1026:0] = A[513:0] x μQ[512:0], wherein μQ[512] = 1;

setting B[513:0] = Z[1026:512];

setting C[513:0] = X[513:0];

calculating Y[1025:0] = B[513:0] x Q[511:0];

setting D[513:0] = Y[513:0];

calculating E[513:0] = C[513:0] - D[513:0];

if $E > Q$ then calculating $E = E - Q$;

if $E > Q$ then $E = E - Q$; and

setting $X_Q = E[511:0]$ as the result of the cleaving X mod Q, whereby X_Q equals X mod Q.

23. (Previously Presented) The method of Claim 18, wherein merging C_P and C_Q to compute C comprises:

if $C_P > P$ then calculating $C_P = C_P - P$;

if $C_Q > Q$ then calculating $C_Q = C_Q - Q$;

calculating $A[512:0] = C_Q[511:0] - C_P[511:0]$;

if $A < 0$ then calculating $A[511:0] = A[511:0] + Q[511:0]$;

calculating $B[1023:0] = A[511:0] \times P^{-1}[511:0]$;

calculating $D[511:0] = \text{Cleave } B[1023:0] \bmod Q[511:0]$, wherein $\mu Q[512] = 1$;

calculating $E[1023:0] = D[511:0] \times P[511:0]$;

calculating $C[1023:0] = E[1023:0] + Cp[511:0]$; and

wherein $C[1023:0]$ is the result of merging C_P and C_Q.

24.(Previously Presented) The apparatus of claim 1, wherein said chaining controller is adapted to implement a flexible chaining algorithm.

25. (Previously Presented) The method of Claim 18, wherein X is data which is to be encrypted.

26. (Previously Presented) The method of Claim 18, wherein X is data which is to be decrypted.

27. (Previously Presented) The method of claim 18, wherein the session controller selected is one 32 available session controllers.

28. (Currently Amended) A device for performing computations, comprising:

a plurality of exponentiators; and

a chaining controller adapted to arrange a first group of said exponentiators into a first computational chain when the device is required to process exponentiations of a first size, and being further adapted to arrange a second group of said exponentiators into a second computational chain when the device is required to process exponentiations of a second size distinct from said first size;

wherein the memories of said plurality of exponentiators map into a single global address.

29. (Previously Presented) The device of claim 28, wherein said plurality of exponentiators operate independently when the device is required to process exponentiations of a third size distinct from said first and second sizes.

30. (Previously Presented) The device of claim 28, wherein said third size is 1K.

31. (Previously Presented) The device of claim 28, wherein said first size is 4K.

32. (Previously Presented) The device of claim 28, wherein said second size is 2K.

33. (Previously Presented) The device of claim 28, wherein the first and second groups are mutually exclusive.

34. (Previously Presented) The device of claim 28, wherein said device is adapted to perform RSA exponentiations.

35. (Previously Presented) The device of claim 28, wherein each of said plurality of exponentiators is adapted to simultaneously process eight 1K exponentiations, four 2K exponentiations, or two 8K exponentiations.

36. (Previously Presented) The device of claim 28, wherein said chaining controller is a direct memory access controller which is adapted to load arguments and control information into the internal memory and registers of said plurality of exponentiators.

37. (Previously Presented) The device of claim 36, wherein said chaining controller is further adapted to retrieve exponentiation results from the memory of said plurality of exponentiators.

38. (Previously Presented) The device of claim 36, wherein said chaining controller is further adapted to retrieve exponentiation results from the memory of said plurality of exponentiators via a single burst-capable thirty-two bit bus interface.

39. (Previously Presented) The device of claim 28, wherein each of said plurality of exponentiators is independently connected to said chaining controller.

40. (Cancelled) The device of claim 28, wherein the memories of said plurality of exponentiators map into a single global address.

41. (Cancelled) The device of claim 28, wherein the memories of said plurality of exponentiators map into a single global address.

42. (Previously Presented) The device of claim 36, wherein each of said plurality of exponentiators comprises a plurality of session controllers, and wherein each of said plurality of session controllers is adapted to process separate exponentiations concurrently.

43. (Previously Presented) The device of claim 42, wherein each of said plurality of exponentiators comprises a single set of computation hardware shared between the plurality of session controllers in a pipelined manner.

44. (Cancelled) The apparatus of Claim 1, wherein each exponentiator is adapted to perform 1024-bit exponentiation.

45. (Cancelled) The apparatus of Claim 44, wherein said chaining controller is adapted to instruct the first subset of devices to act as a first computational chain when the apparatus is required to perform a 2048-bit exponentiation.

46. (Cancelled) The apparatus of Claim 45, wherein said chaining controller is adapted to instruct the second subset of devices to act as a second computational chain when the apparatus is required to perform a 4096-bit exponentiation.

47. (Currently Amended) The apparatus of claim 1 ~~claim 46~~, wherein the first computational chain comprises two computational devices ~~exponentiators~~, and wherein the second computational chain comprises four computational devices ~~exponentiators~~.

48. (Currently Amended) The apparatus of claim 1 ~~claim 46~~, wherein at least some of the computational devices ~~exponentiators~~ in the first computational chain are also in the second computational chain.

49. (New) A device for performing computations, comprising:

a plurality of exponentiators, wherein each of said plurality of exponentiators is adapted to simultaneously process eight 1K exponentiations, four 2K exponentiations, or two 8K exponentiations; and

a chaining controller adapted to arrange a first group of said exponentiators into a first computational chain when the device is required to process exponentiations of a first size, and being further adapted to arrange a second group of said exponentiators into a second computational chain when the device is required to process exponentiations of a second size distinct from said first size.

50. (New) A device for performing computations, comprising:

a plurality of exponentiators;

a chaining controller adapted to arrange a first group of said exponentiators into a first computational chain when the device is required to process exponentiations of a first size, and being further adapted to arrange a second group of said exponentiators into a second computational chain when the device is required to process exponentiations of a second size distinct from said first size; and

a hardware state controller for each exponentiator of the first exponentiation chain; wherein each hardware state controller includes replicated fanout control logic.

51. (New) A device for performing computations, comprising:

a plurality of exponentiators;

a chaining controller adapted to arrange a first group of said exponentiators into a first computational chain when the device is required to process exponentiations of a first size, and being further adapted to arrange a second group of said exponentiators into a second computational chain when the device is required to process exponentiations of a second size distinct from said first size; and

a cleave/merge engine;

wherein the cleave/merge engine is configured to

(a) receive AA, which is a 2w-bit number,

(b) calculate A_1 and A_2, which are two w-bit numbers based on AA, and

(c) output A_1 and A_2;

(d) receive B_1 and B_2, which are two w-bit numbers,

(e) calculate BB, which is a 2w-bit number based on B_1 and B_2, and

(f) output BB; wherein exponentiation of AA yields BB;

wherein exponentiation of A_1 yields B_1;

wherein exponentiation of A_2 yields B_2; and

wherein w is a positive integer.

52. (New) A device for performing computations, comprising:

a plurality of exponentiators; and

a chaining controller adapted to arrange a first group of said exponentiators into a first computational chain when the device is required to process exponentiations of a first size, and being further adapted to arrange a second group of said exponentiators into a second computational chain when the device is required to process exponentiations of a second size distinct from said first size;

wherein said chaining controller is a direct memory access controller which is adapted to load arguments and control information into the internal memory and registers of said plurality of exponentiators, wherein each of said plurality of exponentiators comprises a plurality of session controllers, and wherein each of said plurality of session controllers is adapted to process separate exponentiations concurrently.